



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used **transform string macro**

Found 803 of 124,998

Sort results
by

Display
results



[Save results to a Binder](#)



[Search Tips](#)

☐ Open results in a new
window

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Syntax macros and extended translation](#)

B. M. Leavenworth

November 1966 **Communications of the ACM**, Volume 9 Issue 11

Full text available: pdf(628.14 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

A translation approach is described which allows one to extend the syntax and semantics of a given high-level base language by the use of a new formalism called a syntax-macro. Syntax-macros define string transformations based on syntactic elements of the base language. Two types of macros are discussed, and examples are given of their use. The conditional generation of macros based on options and alternatives recognized by the scan are also described.

2 [A language independent macro processor](#)

William M. Waite

July 1967 **Communications of the ACM**, Volume 10 Issue 7

Full text available: pdf(1.06 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The problem of obtaining starting values for the Newton-Raphson calculation of \sqrt{x} on a digital computer is considered. It is shown that the conventionally used best uniform approximations to \sqrt{x} do not provide optimal starting values. The problem of obtaining optimal starting values is stated, and several basic results are proved. A table of optimal polynomial starting values is given.

3 [The translation of programming languages through the use of a graph transformation language](#)

Peter van den Bosch

March 1982 **ACM SIGPLAN Notices**, Volume 17 Issue 3

Full text available: pdf(816.80 KB) Additional Information: [full citation](#), [references](#)

4 [BIGMAC II: A FORTRAN language augmentation tool](#)

Eugene W. Myers, Leon J. Osterweil

March 1981 **Proceedings of the 5th international conference on Software engineering**

Full text available: pdf(1.00 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

This paper describes the motivation, design, implementation, and some preliminary performance characteristics of BIGMAC II, a macro definition capability for creating language enhancers and translators. BIGMAC II enables the user to specify transformations through STREX, a FORTRAN-like language, which enables the specification of macros which are then used to interpretively alter incoming programs. BIGMAC II is specially adapted to the processing of FORTRAN programs. This paper shows how it ...

5 MACRO: a programming language

Stephen R. Greenwood

December 1979 **ACM SIGPLAN Notices**, Volume 14 Issue 12

Full text available:  [pdf\(1.41 MB\)](#) Additional Information: [full citation](#), [references](#)

6 Pointcuts and advice in higher-order languages

David B. Tucker, Shriram Krishnamurthi

March 2003 **Proceedings of the 2nd international conference on Aspect-oriented software development**

Full text available:  [pdf\(987.37 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Aspect-oriented software design will need to support languages with first-class and higher-order procedures, such as Python, Perl, ML and Scheme. These language features present both challenges and benefits for aspects. On the one hand, they force the designer to carefully address issues of scope that do not arise in first-order languages. On the other hand, these distinctions of scope make it possible to define a much richer variety of policies than first-order aspect languages permit. In this p ...

7 Stack Machines and Classes of Nonnested Macro Languages

Joost Engelfriet, Erik Meineche Schmidt, Jan van Leeuwen

January 1980 **Journal of the ACM (JACM)**, Volume 27 Issue 1

Full text available:  [pdf\(1.46 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 The mobile programming system: STAGE2

W. M. Waite

July 1970 **Communications of the ACM**, Volume 13 Issue 7

Full text available:  [pdf\(698.40 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

STAGE2 is the second level of a bootstrap sequence which is easily implemented on any computer. It is a flexible, powerful macro processor designed specifically as a tool for constructing machine-independent software. In this paper the features provided by STAGE2 are summarized, and the implementation techniques which have made it possible to have STAGE2 running on a new machine with less than one man-week of effort are discussed. The approach has been successful on over 15 machines of wide ...

Keywords: bootstrapping, implementation techniques, machine independence, macro processing, programming languages

9 A syntax-directed approach to automated aids for symbolic mathematics

Lewis C. Clapp

January 1966 **Proceedings of the first ACM symposium on Symbolic and algebraic manipulation**

Full text available:  [pdf\(789.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Designing a system for the analytic processing of mathematical functions around a central syntax processor and permitting the user to work within the context of mathematical syntax leads to a flexible system with a broad range of capabilities. One advantage to this approach is that the basic system can be developed without many a priori restrictions on the nature of the mathematical entities to be processed. Once the basic structure has been developed, the user is free to define and operate ...

10 Algorithm 803: a simpler macro processor

William A. Ward

June 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 2

Full text available:  [pdf\(73.41 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Macro processors have been in the computing tool chest since the late 1950's. Their use, though perhaps not what it was in the heyday of assembly language programming, is still widespread. In the past, producing a full-featured macro processor has required significant effort, similar to that required to implement the front-end to a compiler augmented by appropriate text substitution capabilities. The tool described here adopts a different approach. The text containing macro definitions and ...

Keywords: awk, portable, simple

11 Context-free grammars on trees

William C. Rounds

May 1969 **Proceedings of the first annual ACM symposium on Theory of computing**


Full text available:  [pdf\(358.99 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this paper we discuss still another version of indexed grammars 1 and macro grammars3, gaining some geometric intuition about the structure of these systems. An ordinary context-free grammar is a rewriting system for strings; we find that a macro grammar is a rewriting system for trees. CF grammars on strings form a special case since strings can be thought of as trees without branching nodes. We consider the special case of finite-stat ...

12 The role of programming in a Ph.D. computer science program

Bruce W. Arden

January 1969 **Communications of the ACM**, Volume 12 Issue 1

Full text available:  [pdf\(1.01 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this general paper the role of programming in advanced graduate training is discussed. Subject matter related to programming as well as programming per se is considered. The importance and application of formalism are considered and also the need for good empirical experimentation. A brief outline for a sequence of courses is included, and subject headings that have been obtained from an extensive bibliography are given. A bibliography of programming references is included.

Keywords: course content, course sequence, graduate curriculum, graduate programs, graduate-level programming, programming bibliography, programming research topics

13 Syntax translation with context macros or macros without arguments

Charles A. Grant

September 1971 **ACM SIGPLAN Notices**, **Proceedings of the international symposium on**

Full text available:  pdf(149.20 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A conceptually language independent macro facility with the following properties is described here: 1) Each macro gathers its own arguments from the text in which the call is embedded (the context). 2) A macro may modify its context an arbitrary distance from the actual string which represents the call. 3) A macro may specify the precise point from which scanning is to continue after the macro returns. The objective of this work is to define a simp ...

14 BASIC Zgrass—a sophisticated graphics language for the Bally Home Library Computer

Tom DeFanti, Jay Fenton, Nola Donato

August 1978 **Proceedings of the 5th annual conference on Computer graphics and interactive techniques**

Full text available:  pdf(477.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Home computer users are just now discovering computer graphics. Modest extensions to BASIC allow plotting but not much more. The Bally Home Library Computer, however, has hardware to aid implementation of video games. Custom integrated circuits working on a 160×102 pixel (2 bits per pixel) color television screen allow certain forms of animation in real time. To give this power to the user, BASIC Zgrass has been designed and implemented. It is an extension of BASIC that allows paralle ...

Keywords: Art, Graphic language, Interpreters, Interactive computer graphics, Real-time

15 Information Retrieval: Easy English, a language for information retrieval through a remote typewriter console

M. Rubinoff, S. Bergman, F. Rapp, H. Cautin

October 1968 **Communications of the ACM**, Volume 11 Issue 10

Full text available:  pdf(475.02 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)


Easy English is a natural command language designed to simplify communication between man and machine through remote typewriter console. It has been developed for retrieval of documents from a computerized data base, the Moore School Information Systems Laboratory files. Requests are formulated in a standardized syntactical form (examples of which are presented), and this form is then transformed into an equivalent query expressed in the retrieval system's original Symbolic Command Language ...

Keywords: conversational mode, document retrieval, information retrieval, information retrieval language, man-machine communication, natural language communication, on-line searching, remote console communication, remote terminal communication, symbolic command language, translator

16 The AED approach to generalized computer-aided design

Douglas T. Ross

January 1967 **Proceedings of the 1967 22nd national conference**

Full text available:  pdf(2.02 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper has been written in response to a request for an up-to-date broad view of the approach to computer-aided design taken by the M.I.T. Computer-Aided Design Project. Included in the suggestion was the hope that such a description would help to illuminate, especially for people who are not system programmers, the major features which any computer-aided design system must have in order to be a useful practical tool. This has

proved to be a difficult assignment, because there are sever ...

17 A history of the SNOBOL programming languages

Ralph E. Griswold

January 1978 **The first ACM SIGPLAN conference on History of programming languages**


Full text available:  [pdf\(3.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Development of the SNOBOL language began in 1962. It was followed by SNOBOL2, SNOBOL3, and SNOBOL4. Except for SNOBOL2 and SNOBOL3 (which were closely related), the others differ substantially and hence are more properly considered separate languages than versions of one language. In this paper historical emphasis is placed on the original language, SNOBOL, although important aspects of the subsequent languages are covered.

18 Code migration through transformations: an experience report

K. Kontogiannis, J. Martin, K. Wong, R. Gregory, H. Müller, J. Mylopoulos

November 1998 **Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(203.77 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

One approach to dealing with spiraling maintenance costs, manpower shortages and frequent breakdowns for legacy code is to "migrate" the code into a new platform and/or programming language. The objective of this paper is to explore the feasibility of semiautomating such a migration process in the presence of performance and other constraints for the migrant code. In particular, the paper reports on an experiment involving a medium-size software system written in PL/IX. Several modules of the sy ...

19 Session 1 (full technical papers): evolution in source code: Challenges of refactoring C programs

Alejandra Garrido, Ralph Johnson

May 2002 **Proceedings of the international workshop on Principles of software evolution**

Full text available:  [pdf\(687.83 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Refactoring has become a well-known technique for transforming code in a way that preserves behavior. Refactorings may be applied manually, although manual code manipulation is error prone and cumbersome, so maintainers need tools to make automatic refactorings. There is currently extensive literature on refactoring object-oriented programs and some very good tools for refactoring Smalltalk and Java code. Although there is more code written in C or C++ than in any other language, refactoring too ...

Keywords: C programming, preprocessor directives, refactoring

20 Programmable formatting of program text: Experiences drawn from the TAMPR system

Kenneth W. Dritz

October 1978 **Proceedings of the SIGNUM Conference on the Programming Environment for Development of Numerical Software**

Full text available:  [pdf\(155.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The TAMPR System originated as an approach to the problem of automating the routine modifications of Fortran source programs required to adapt them to a variety of uses or environments [1]**. Overall, the system accomplishes such modifications by applying transformations to Fortran programs at the source level. But the process differs markedly, in detail, from string-based editing or macro expansion. Three steps are involved: (1) A Fortran source program is processed by the TAMPR ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Searching for PHRASE transform string macro.

Restrict to: [Header](#) [Title](#) Order by: [Citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Amazon](#) [B&N](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

No documents match Boolean query. Trying non-Boolean relevance query.

1000 documents found. Only retrieving 250 documents (System busy - maximum reduced). Retrieving documents... Order: relevance to query.

[A More Precise Solution to Two Problems on Tries - Navarro, Poblete \(Correct\)](#)

V. Poblete Abstract We use the Binomial Transform to address the problem of determining the of A [2]The trie is intended to store a set of strings over A, and retrieve any of them in time
ftp.dcc.uchile.cl/pub/users/gnavarro/tries.ps.gz

[Initialization Halting Concepts and Their Basic Properties of.. - Jingchao Chen \(Correct\)](#)

380 Japan Summary. Up to now, many properties of macro instructions of SCMFSA are described by the InitHalting basic properties of the compositions of macro instructions,if-Macro(conditional branch macro of the compositions of macro instructions,if-Macro(conditional branch macro instructions) and mizar.org/JFM/Vol10/..ps/scm_halt.ps

[A Macro Expansion Approach to Embedded Processor Code Generation - Lassila \(1996\) \(Correct\)](#)

flow analysis. The main advantage of the tool is its strong support for macro hierarchy: hierarchical macro A Macro Expansion Approach to Embedded Processor Code
The tool is a retargetable assembly-code-level macro expander capable of program flow analysis. The saturn.tcs.hut.fi/pub/reflex/euromicro96.ps.Z

[Macro-Actions in Reinforcement Learning: An Empirical Analysis - McGovern, Sutton \(1998\) \(Correct\)](#)

at the level of the most primitive actions. One strategy for overcoming this difficulty is to combine Amherst Technical Report Number 98-70 1 Macro-Actions in Reinforcement Learning: An Empirical learning by using temporally extended actions, or macro-actions, but none has carefully analyzed what www-anw.cs.umass.edu/~amy/pubs/mcgovern-techrpt-98-70.ps

[Refining Hygienic Macros for Modules and Separate Compilation - Blume \(Correct\)](#)

Q QIde qualified identifiers T Tf macro transformers M Mod modules E Exp expressions of Computer Science, Princeton University 35 Olden Street, Princeton, NJ 08544 e-mail: Refining Hygienic Macros for Modules and Separate Compilation Matthias
www.kurims.kyoto-u.ac.jp/~blume/papers/hygmac.ps.gz

[Macros in Scheme - Clinger \(1991\) \(Correct\) \(1 citation\)](#)

is a general mechanism for defining syntactic transformations that reliably obey the rules of lexical than Lisp, Scheme has just become the first block-structured programming language to support a reliable Macros in Scheme William Clinger Although macros are
ftp.cs.indiana.edu/pub/scheme-repository/doc/prop/macinsch.ps.gz

[Pkcs #5: Password-Based Encryption Standard - Versions \(1993\) \(Correct\)](#)

encryption algorithm. Both processes transform an octet string to another octet string. The Pbeparameter :Sequence {Salt Octet String Size(8)Iterationcount Integer }The Fields Of
www.dice.ucl.ac.be/crypto/doc/PKCS/pkcs-5.ps.gz

[Writing Hygienic Macros in Scheme with Syntax-Case - Dybvig \(1992\) \(Correct\) \(5 citations\)](#)

used Scheme hdatum representations. Macro transformers take syntax objects as input and return information about an expression in addition to its structure. This contextual information enables the Writing Hygienic Macros in Scheme with Syntax-Case R. Kent Dybvig
ftp.cs.indiana.edu/indra/scheme-repository/doc/prop/syntax-case.ps.gz

[Macro Control Languages And Actions For Cocolog - Martínez-Mascarua, CAINES \(1996\) \(Correct\)](#)

the CAs themselves are embedded in a language-like structure consisting of sets of CCRs. This enhancement Of Control And Optimization September, 1996. Macro Control Languages And Actions For Cocolog C. Mart complex predicates and functions in terms of simple Macro COCOLOG Language symbols which can be defined

<ftp.cim.mcgill.ca/pub/papers/1996/bigmac.ps.gz>

Babel, a multilingual package for use with LATEX's standard.. - Braams (1995) (Correct)

7.25 "else 7.26 "errmessage-No room for a new "string"language!7.27 "fi 7.28
to another he can do so using "selectlanguage the macro "selectlanguage. This macro takes the language,
"selectlanguage the macro "selectlanguage. This macro takes the language, defined previously by a
www.math.fu-berlin.de/~fuchs/babel.ps

A Data Transformation System for Biological Data Sources - Buneman, Davidson, Hart, .. (1995) (Correct)
(32 citations)

A Data Transformation System for Biological Data Sources P.
resides not only in conventional databases, but in structured files maintained in a number of different
sdmc.krdl.org.sg/kleisli/psZ/bdkowvldb95.ps

ForML - a Pretty-Printing Facility for SML - Rohwedder (1993) (Correct)

Unparsing and Formatting Lexer and parser transform input text into trees containing the abstract
Oppen [3]the ForML formatting is not based on a stream interface: an intermediate formatting structure
www.ii.uni.wroc.pl/~tomasz/sml/doc/tools/pretty.ps.gz

A temporal mode selection in the MPEG-2 encoder scheme - Piron (Correct)

2) standard [2] defines only the syntax of the bit-stream. There is no constraint on the encoder. This
Each frame is divided into smaller blocks, called macro-blocks, of size 16\Theta16 pixels. A macro-block
called macro-blocks, of size 16\Theta16 pixels. A macro-block may be Intra-coded (without any temporal
lts3g3.epfl.ch/text/publications/.../publications/papers/lp_eusipco96.ps.gz

Syntactic Abstraction in Scheme - Hieb, Dybvig, Bruggeman (1992) (Correct) (5 citations)

(Revised July 3, 1992) Abstract Naive program transformations can have surprising effects due to the
problem for variables and constants as well as structured expressions, ffl supplies a
Introduction A fundamental problem with most Lisp macro systems is that they do not respect lexical
ftp.cs.indiana.edu/pub/scheme-repository/doc/pubs/iucstr355.ps.gz

Hygienic Macros Through Explicit Renaming - Clinger (1991) (Correct) (4 citations)

other hygienic macros. The problem is that the transformation procedure for the defined macro may need
Hygienic Macros Through Explicit Renaming William Clinger This
paper describes an alternative to the low-level macro facility described in the Revised 4 Report on
ftp.cs.indiana.edu/pub/scheme-repository/doc/prop/exrename.ps.gz

Building an HTML macro toolbox - Andrew Peel (1996) (Correct)

with a description of Rummage, a tool for inserting structure-related macros into collections of web nodes
Ukc Computing March 1996 Building An Html Macro Toolbox Building An Html Macro Toolbox Andrew
Building an HTML macro toolbox Building an HTML macro toolbox Andrew Peel (A.T.Peel@ukc.ac.uk)
www.cs.ukc.ac.uk/pubs/1996/3/content.ps.gz

Optimal Parallel Algorithms for Periods, Palindromes and.. - Apostolico, al. (1992) (Correct) (1 citation)

are presented: ffl Finding all the periods of a string. The period of a string can be computed by
www.cs.columbia.edu/~dany/papers/ppsicalp.ps.Z

Solitonic Strings and BPS Saturated Dyonic Black Holes - Cvetič, Tseytlin (1995) (Correct) (2 citations)

When acted by combined T- and S-duality transformations it serves as a generating solution for
hep-th/9512031 December 1995 Solitonic Strings and BPS Saturated Dyonic Black Holes Mirjam
preprints.cern.ch/archive/electronic/hep-th/9512/9512031.ps.gz

Partial Evaluation of Pattern Matching in Strings - Consel, Danvy (1989) (Correct) (20 citations)

evaluation is a semantics-preserving program transformation based on propagating constants, unfolding
Ed. Partial Evaluation Of Pattern Matching In Strings Charles Consel Olivier Danvy Litp Diku -
ftp.daimi.aau.dk/pub/empl/danvy/Papers/consel-danvy-ipl89.ps.gz

acquire-macros: An Algorithm for Automatically Learning.. - McGovern (1998) (Correct) (1 citation)

Digney, B. 1998) Learning hierarchical control structure for multiple tasks and changing
and Hierarchy in Reinforcement Learning 1 acquire-macros: An Algorithm for Automatically Learning
An Algorithm for Automatically Learning Macr -actions Amy McGovern amy@cs.umass.edu Computer

www-anw.cs.umass.edu/~amy/pubs/mcgovern_nips98_workshop.ps.gz

First 20 documents [Next 20](#)

Try your query at: [Amazon](#) [Barnes & Noble](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - citeseer.org - [Terms of Service](#) - [Privacy Policy](#) - Copyright © 1997-2002 [NEC Research Institute](#)

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

 Print Format
Your search matched **3** of **990765** documents.A maximum of **3** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.

You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

(string)and (macro)

Search Again

Results:

Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**1 **Splicing systems on graphs***Freund, R.;*

Intelligence in Neural and Biological Systems, 1995. INBS'95, Proceedings., First International Symposium on , 29-31 May 1995

Page(s): 189 -194

[\[Abstract\]](#) [\[PDF Full-Text \(408 KB\)\]](#) **IEEE CNF**
2 **An instrument control and data analysis program configured for NMR imaging***Roos, M.S.; Mushlin, R.A.; Veklerov, E.; Port, J.D.; Ladd, C.; Harrison, C.G.;*
Nuclear Science, IEEE Transactions on , Volume: 36 Issue: 1 , Feb. 1989

Page(s): 988 -992

[\[Abstract\]](#) [\[PDF Full-Text \(476 KB\)\]](#) **IEEE JNL**
3 **Hybrid genetic algorithms for constrained placement problems***Schnecke, V.; Vornberger, O.;*

Evolutionary Computation, IEEE Transactions on , Volume: 1 Issue: 4 , Nov. 1997

Page(s): 266 -277

[\[Abstract\]](#) [\[PDF Full-Text \(280 KB\)\]](#) **IEEE JNL**